

技术方案设计

目 录

- 1、设计概述
 - 1.1功能背景
 - 1.2非功能约束
- 2、设计思路及折衷
- 3、系统设计
 - 3.1系统部署图
 - 3.2下单抢单场景时序图
 - 3.3订单状态图
- 4、订单交易模块详细设计
- 5、订单调度模块详细设计
- 6、订单履约模块详细设计
- 7、性能设计
- 8、容灾设计
- 9、开发计划
- 10、上线方案

1、设计概述

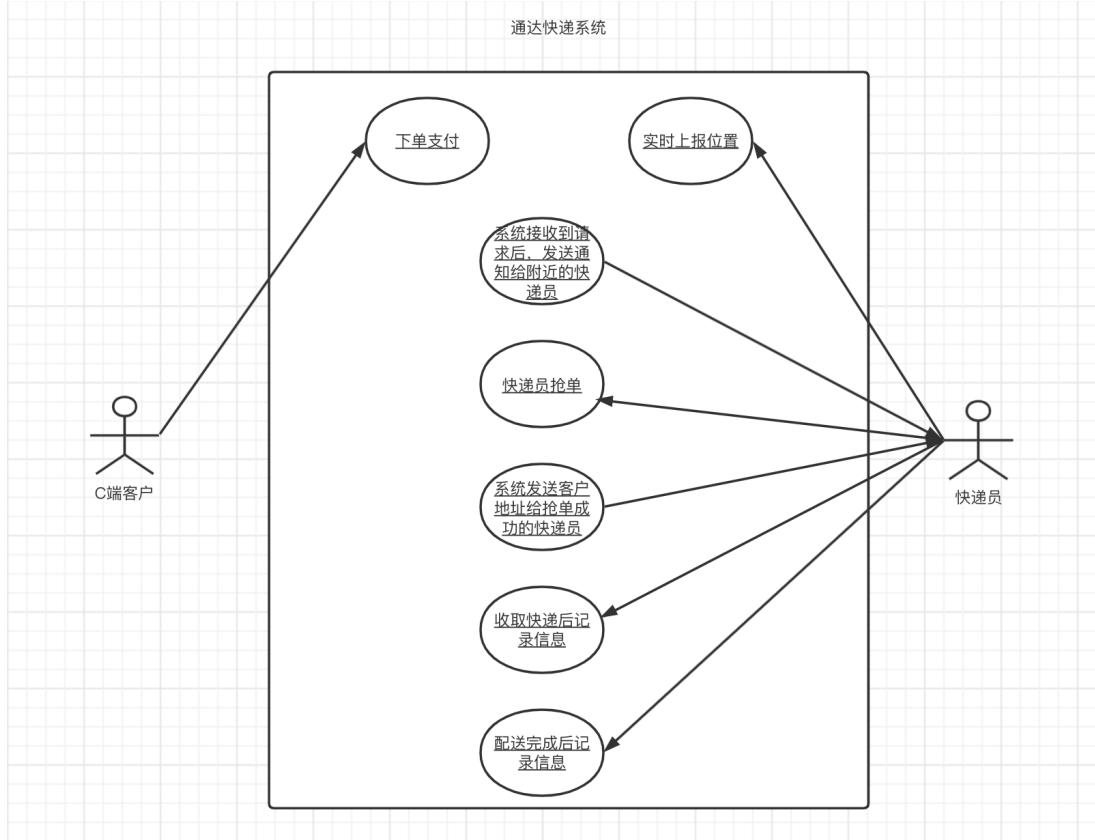
通达快递系统是一个直接面向C端用户的互联网系统，是公司互联网转型战略的核心系统，承担着公司业务倍增的目标任务。

1.1功能背景

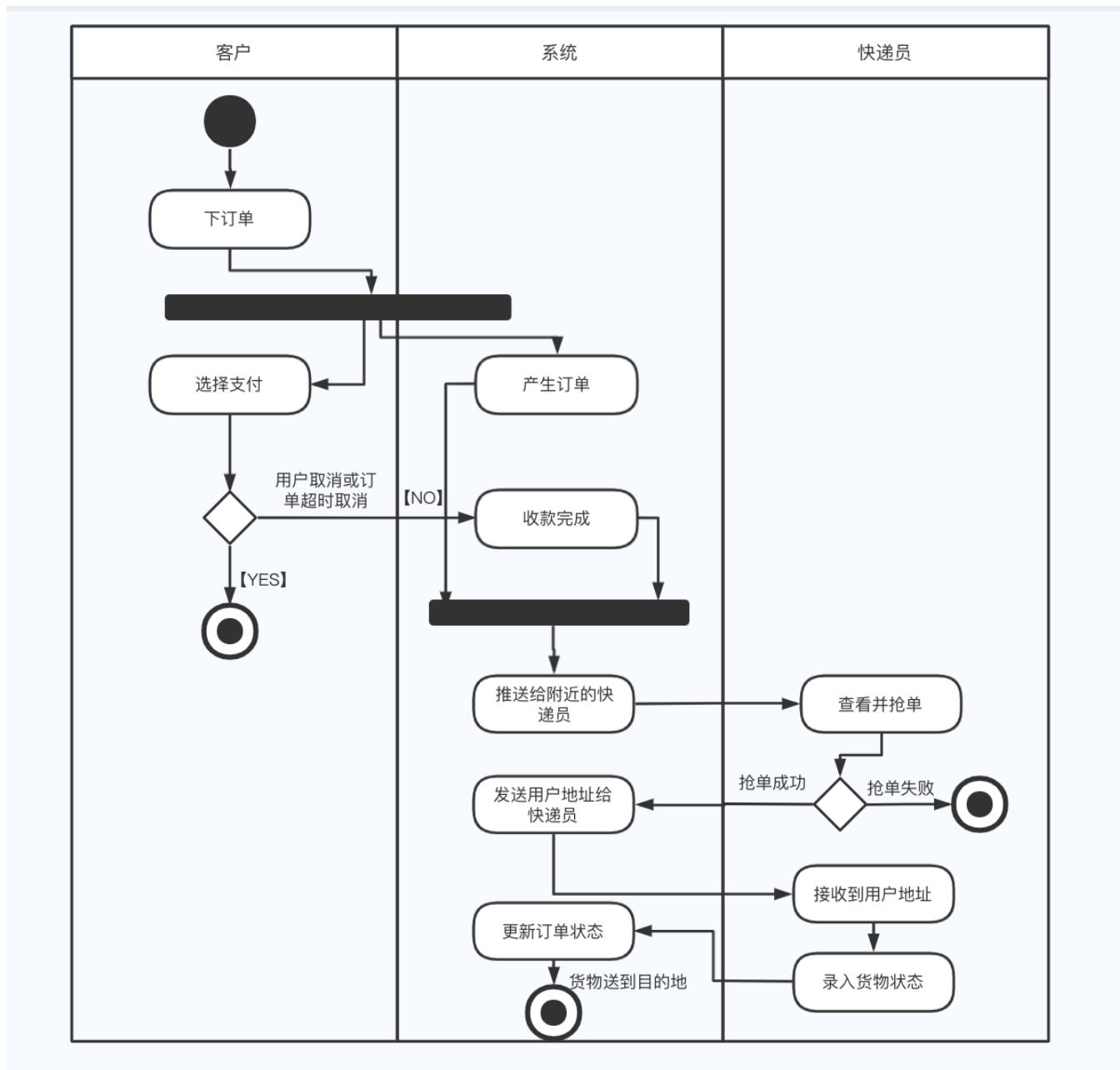
系统使用者包括有快递需求的C端客户和公司快递员，系统主要功能如下所示：

- 1、用户通过app发起快递下单请求并支付
- 2、快递员通过自己的App上报自己的地理位置，每30秒上报一次
- 3、系统收到快递请求后，向距离用户直线距离5km内的所有快递员发送通知
- 4、快递员需要进行抢单，第一个抢单的快递员得到配单，系统向其发送用户详细地址
- 5、快递员到用户处收取快递，并记录到系统中：已收件
- 6、快递员将快递送到目的地，并记录到系统中：已送达

核心用例图如下：



核心业务活动图如下：



1.2 非功能约束

预计系统上线后三个月日单超过1万，一年日单超过50万，核心指标如下：

- 1、查询性能目标：平均响应时间<300ms, 95%响应时间<500ms, 单机TPS>100;
- 2、下单性能目标：平均响应时间<800ms, 95%响应时间<1000ms, 单机TPS>30;
- 3、系统核心功能可用性目标：>99. 99%;
- 4、系统安全性目标：系统可拦截常规网络攻击，密码数据散列加密，客户端数据HTTPS加密，外部系统间通信对称加密；
- 5、数据持久化目标：>99. 9999%。

2、设计思路及折衷

根据三个月及一年内日单量，通达快递系统需要解决和考虑的关键问题如下所示：

关键问题一：下单支付的高可用

在整个系统中，下单支付的场景是最重要的，所以可用性指标需要达到99. 99%，假设订单的分布是全国范围内的，可以做异地多活的多个集群，通过负载均衡分发给下单网关集群

关键问题二：快递员实时位置的读写高性能

假设每个快递员每天可以处理10单，日单超过50万的情况下，需要5W左右快递员，快递员每30秒上传一次位置，意味着30秒的写入超过5万条；同时位置信息的时效比较短，并不需要持久保留；所以为了达到读写的高性能，可以选择Redis来存储；Redis不可用的情况下，可以降级为通过MySQL查询，DB上存储快递员的初始POI信息。

关键问题三：下单支付、派单和抢单的高并发

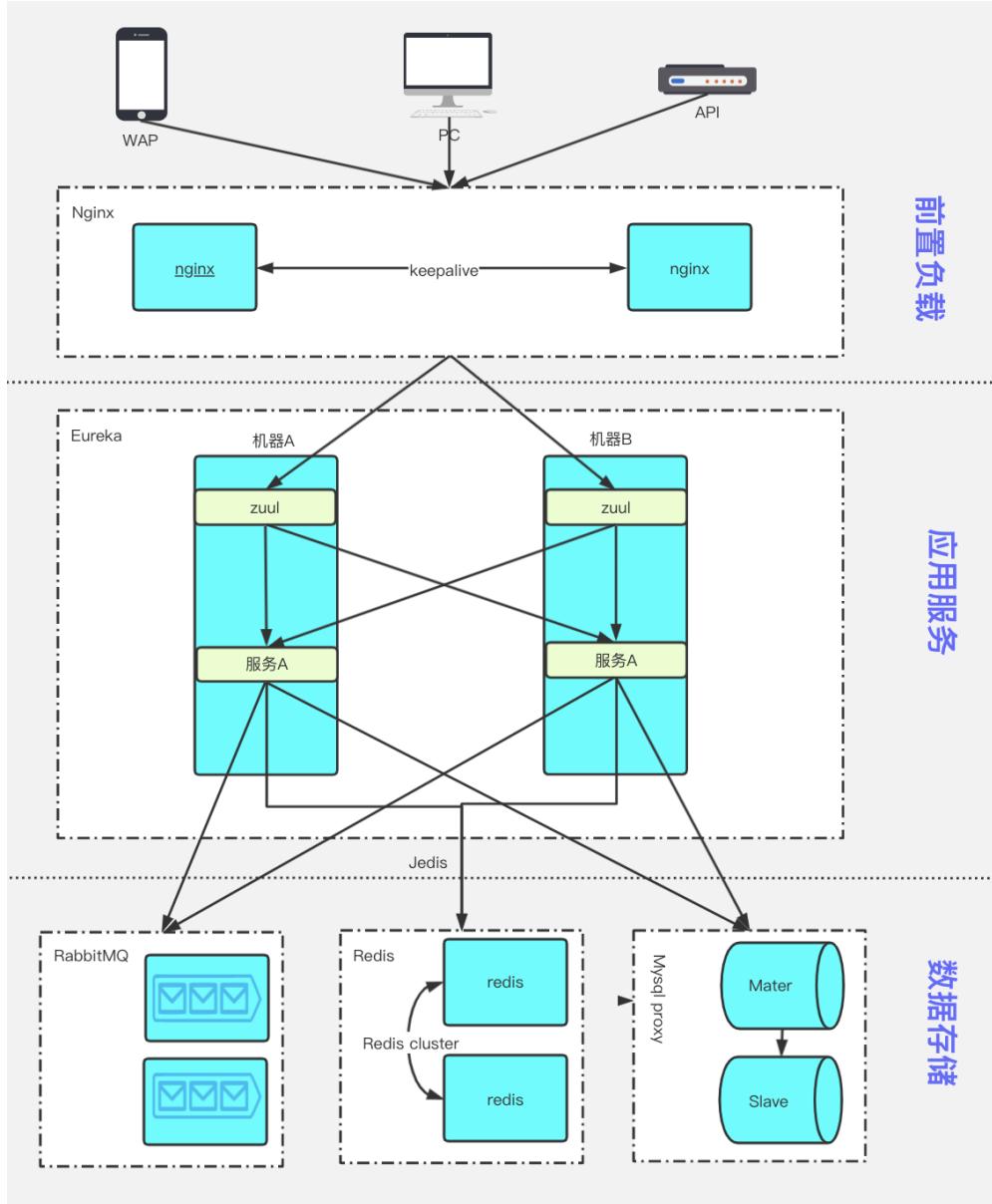
支付完成之后，可以通过消息队列解耦，提升下单支付的接口性能；通过水平扩展消费者服务，可以显著提升派单的处理能力，增强用户体验

关键问题四：海量订单数据的存储问题

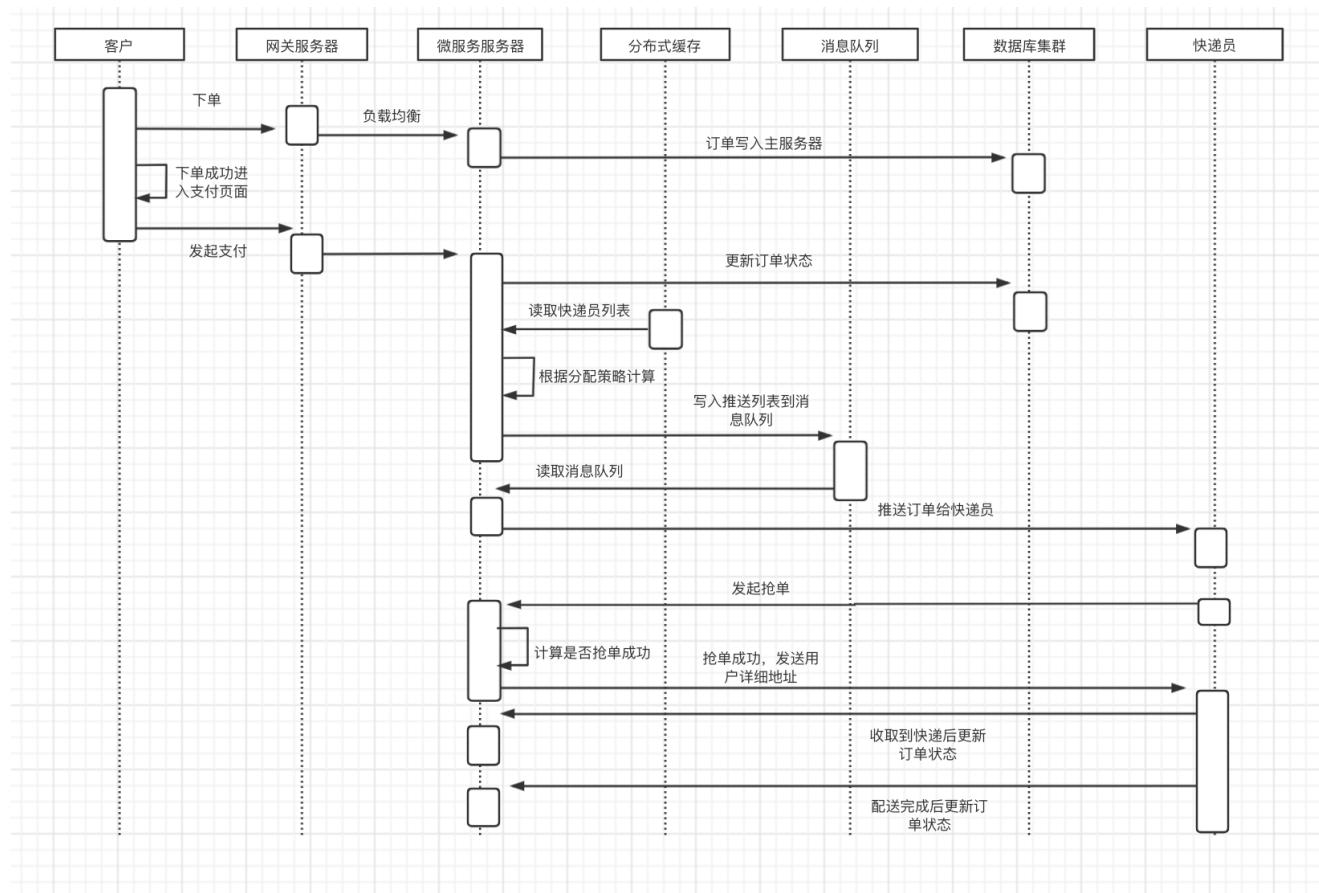
按目前的增长预估，订单量在三个月后可以达到100W以内，一年内肯定会超过1000W，前期可以做一主两从的数据集群，但是要做好分库分表的解决方案

3、系统设计

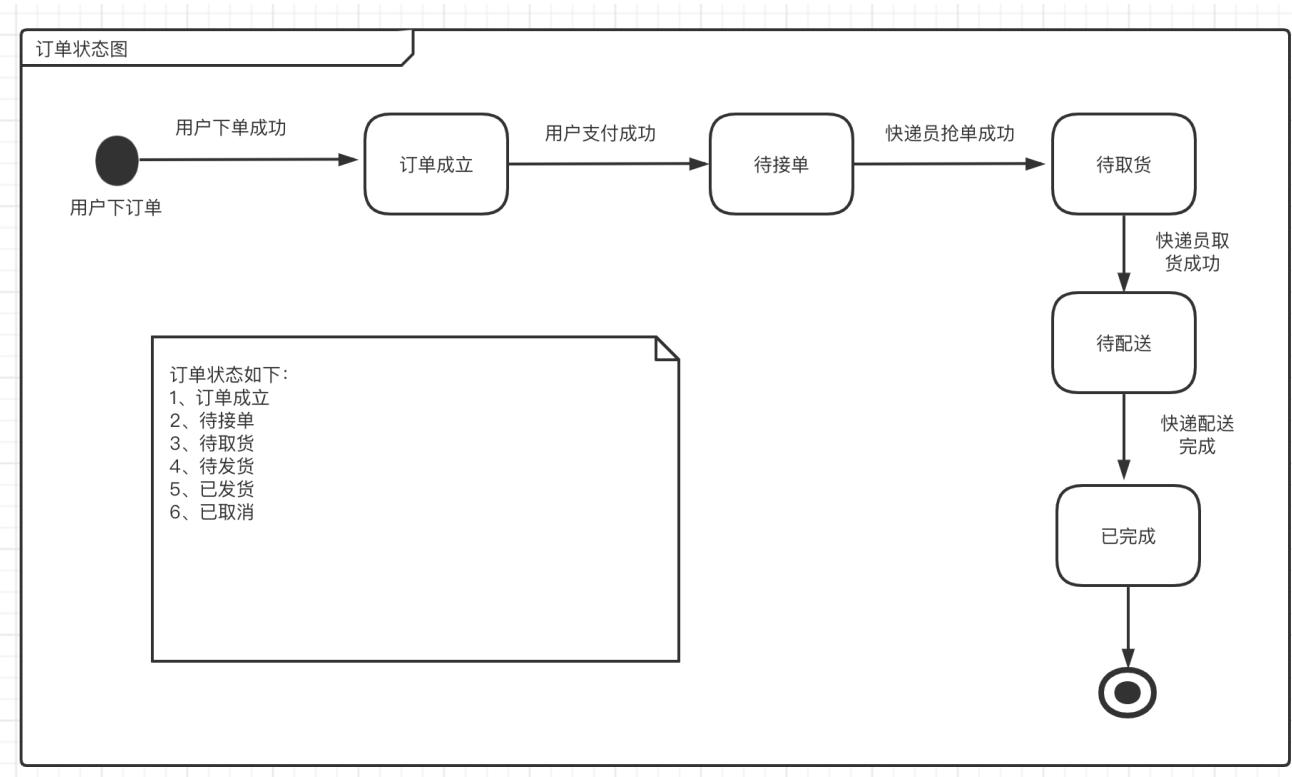
3.1 系统部署图



3.2 下单抢单场景时序图



3.3 订单状态图



4、订单交易模块详细设计

该节描述此模块的DB设计、接口设计、核心流程

5、订单调度模块详细设计

该节描述此模块的DB设计、接口设计、核心流程

6、订单履约模块详细设计

该节描述此模块的DB设计、接口设计、核心流程

7、性能设计

该节描述此项目需要达到的性能指标，以及达到此性能而采用的算法、设计思想、策略。此节不必复述各模块的交互及其功能，只需突出性能考虑即可

8、容灾设计

此节描述为了保证系统的安全、稳定运行，对于数据丢失、服务器无法连接等运行中可能发生的问题，所采取的容灾、保护策略

9、开发计划

此节描述为了将开发周期以及计划信息同步出来，以便于所有人查看，如果是单独连接，可以连接进来

10、上线方案

该节描述此项目的上线步骤，确保上线过程的稳定性